Dis papa (ou maman), comment arrivent les bugs dans le monde numérique ?

Aurélien Alvarez, Thierry Viéville

27 janvier 2014 **1 commentaire — commenter cet article**

Objet du mois

Dis papa (ou maman), comment arrivent les bugs dans le monde numérique ?

Aurélien Alvarez et **Thierry Viéville**

En fait, cela veut dire « insecte rampant », un truc qui tape sur les nerfs, quoi ! Les électriciens de la fin du XXe siècle utilisaient déjà le terme pour parler d'un dysfonctionnement. Et puis, depuis Grace, Grace Murray Hopper qui en a popularisé l'usage, nous voilà toutes et tous en train de l'utiliser en informatique et bien au delas, euh, au delah, ah non, au-delà, excusez-nous, on vient justement d'en avoir un. De « bug » !

Pour comprendre ce qui sous-tend ces « bugs » il faut se souvenir que tout ce qui est dans un ordinateur est affaire de codage : codage binaire au niveau de la machine, langage formel pour coder les instructions du programme, etc.

Nous allons aborder cet élément par **plusieurs facettes** très différentes pour en avoir une vision globale.

Ainsi :

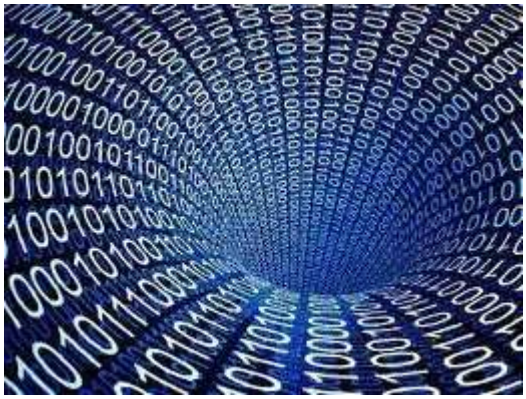
> « Un ordinateur est bourré de circuits électroniques avec de microscopiques interrupteurs pour laisser passer ou pas le courant, ce qui donne des 0 ou des 1, selon que l'interrupteur est ouvert ou fermé.
>
> Programmer en « langage machine », c'est spécifier une à une chaque ouverture ou fermeture des circuits électroniques de la machine. Les premiers ordinateurs n'étaient donc quasiment pas utilisables à grande échelle puisque programmés ainsi. **Grace Murray Hopper** fit sauter ce verrou et défendit l'idée qu'un programme doit pouvoir être écrit dans un langage formel proche de l'anglais. Elle conçut alors un compilateur, c'est-à-dire un logiciel qui traduit en langage machine les éléments de l'algorithme donné dans un langage compréhensible par tous les ingénieurs. »

S'il y a une erreur lors du codage de l'information ou des instructions à exécuter, comme la machine ne fait

qu'exécuter le code, et ne connaît que le code, eh bien elle fera facilement n'importe quoi, avec ce code erroné.

Python, un langage de programmation très très proche de l'anglais

Ce qui est remarquable c'est que tous les nombres, les textes, les sons, les images, les informations ont un reflet numérique, un codage. Avec un bit, on va par exemple coder une réponse « oui » ou « non », disons 0 ou 1. Avec deux bits qui prennent chacun la valeur 0 ou 1, on dispose de quatre codes 00, 01, 10, 11 et on peut coder ainsi quatre éléments, quatre couleurs ou quatre lettres, ou les nombres de 0 à 3 inclus. Avec plus de bits on codera des nombres plus grands, toutes les lettres, etc.



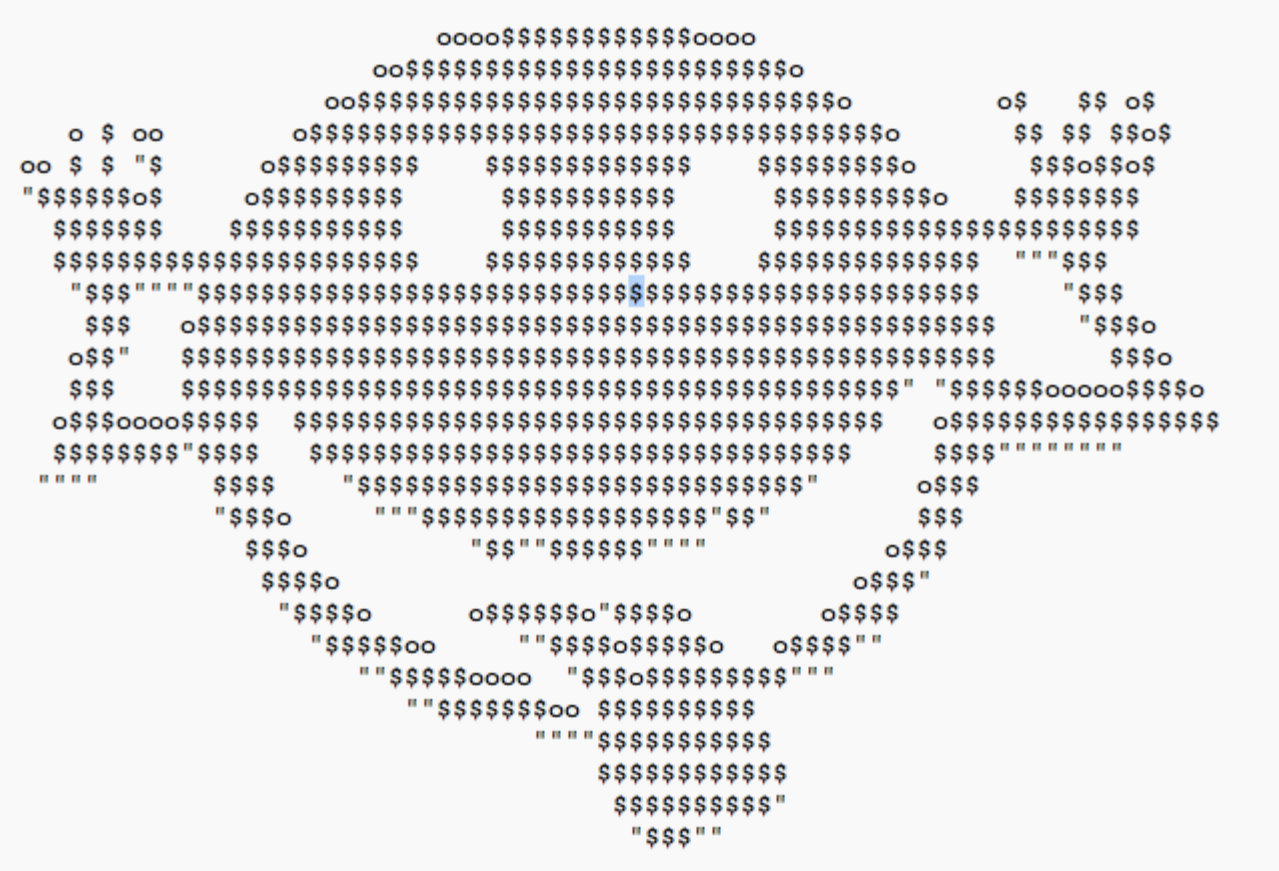
Ce qui est formidable, c'est qu'une fois réduits ces objets à leur code numérique, eh bien toute opération qui consiste à recopier ces objets, les envoyer par mail, les stocker, etc. fonctionne quel que soit le contenu : envoyer une image ou un tableau de nombres utilise les mêmes mécanismes.

Mais comment expliquer cela à nos enfants ?

On pourra par exemple proposer un langage pour coder un tout petit dessin. Au niveau binaire, il faut coder chaque pixel. L'idée est que c'est amusant de donner un code 11111000111000011111 qui, une fois « décodé en 2D », donne un digit :

```
1111
1000
1110
0001
1111
```

Ici une sorte de « S ». Dans ce domaine, il y a de vrais artistes, on parle d'ailleurs de l' **art ASCII** !



Alors deux enfants ou groupes d'enfants se mettent de part et d'autre d'un paravent. On convient qu'ils vont s'échanger de tout petits dessins sur une grille de taille 4 x 5 par exemple. On y dessine des points (disons un 1) ou pas de point (disons un 0) et cela fait des chiffres ou des lettres. Ensuite on transmet à la queue leu-leu les bits et l'autre groupe dessine le résultat et le dessin ré-apparaît.

Mmm... sauf s'il y a eu un bug ! Si par exemple on oublie de transmettre un des bits ça ne fait pas juste une erreur, c'est toute l'image qui se décale et ça donne n'importe quoi. On pourra même s'amuser à envoyer les bits en **verlan**, du dernier au premier, et voir ce qui se passe pour le dessin.

On va maintenant jouer à un tout autre jeu, afin de continuer de découvrir les différentes facettes du codage. Voici huit bonhommes [**1**]. Il faut envoyer un code pour deviner lequel a été choisi.

On utilise trois bits :

- moustache 1 ou pas 0 ?
- lunettes 1 ou pas 0 ?
- chapeau 1 ou pas 0 ? [**2**]

000	001	010	

			
100	101	110	
			

De chaque côté du paravent on aura les huit bonhommes et le jeu sera de deviner quel est le bonhomme choisi par un groupe qui a juste communiqué le code. Évidemment il faut bien se mettre d'accord sur le fait d'envoyer d'abord le bit de moustache, puis celui des lunettes, puis celui du chapeau. Sans cela, ça ne marche pas ! On pourra d'ailleurs, selon le groupe d'enfants, soit leur donner une consigne précise à ce sujet, soit les laisser découvrir cette nécessité.

On pourrait même rajouter des bits supplémentaires d'information (nœud papillon ou non, tirer ou pas la langue) pour détecter si une information est erronée, voire même permettre de trouver le bonhomme en choisissant le personnage qui est le plus proche du code proposé… [stop] ! Il est sûrement l'heure du goûter :-).

Passer du jeu à la séance de cinéma

Allez, il est temps de se reposer et de regarder (en cliquant par exemple sur l'image ci-dessous) un petit **film** de trois minutes [**3**] :

Voici son contenu :

« Pour qu'un ordinateur fasse un truc tout seul, un truc simple comme une addition, ou un truc compliqué comme piloter un vaisseau, faut lui expliquer dans les moindres détails tout ce qu'il doit faire. Mais les terriens [...] savent programmer des machines pour qu'elles fassent des trucs toutes seules et qu'elles ne se trompent jamais [par rapport à ce qu'on leur a demandé].

Un ordinateur ça parle en chiffres… binaires. C'est pas très pratique quand même… Et les humains, ils parlent avec des mots. Les compilateurs, ils traduisent les mots en chiffres ! Comme ça, on peut plus facilement trouver les erreurs et les réparer, grâce à Grace Hopper.

Un jour, son ordinateur est tombé en panne... un tout petit insecte est venu se brûler sur les circuits de ce gros ordinateur ; alors, pour rigoler, Grace a dit que c'est à cause de lui que plus rien ne marche. Comme « insecte » se dit « bug » en anglais, quand un ordinateur fait n'importe quoi, on dit qu'il bugge ! »

[4]

Vous venez de partager la première leçon de codage informatique avec votre enfant de 6-12 ans et votre deuxième leçon d'informatique théorique avec lui si vous avez lu notre **premier article**.

Que venons-nous d'apprendre ensemble ici ?

Tout d'abord on concrétise ici cette idée, souvent bien vague, que « les objets sont codés en binaire dans les ordinateurs ». On voit d'abord que ce codage est un choix, une *convention entre les individus*, exactement comme le langage. On expérimente qu'il est possible de coder toute sorte d'objets, en fait tous les objets que nous trouvons sur Internet ou dans nos smartphones.

Ce qui est intéressant pour l'enfant c'est que cela va l'aider à faire la différence entre le réel et le virtuel. Le codage d'un son ou d'une scène visuelle n'est que le reflet numérique de cet objet réel. Il y a le « S » que je dessine avec de la peinture, il est fait de matière. Il y a ensuite le codage du « S », ce paquet de 0 et de 1, qui ne représente le « S » que parce qu'on le veut bien.

C'est important d'aller assez loin sur cette métaphore et de montrer que, par rapport à une rencontre réelle entre deux amis, la communication à travers un réseau social n'est qu'un échange de codes informatiques. On ne peut pas être « bien ensemble » sur un réseau social, on ne peut que « se figurer être bien ensemble ». Une personne que nous ne connaissons que par son profil d'un réseau social n'existe finalement qu'à travers les informations qui sont partagées. Le « reste » c'est nous qui l'avons imaginé. Ce peut être une expérience très intéressante, en soi, mais qui ne met en jeu qu'une partie de deux êtres qui communiquent ainsi. Cette expérience est donc par définition d'une toute autre nature qu'une véritable rencontre réelle entre deux amis qui peuvent véritablement éprouver, expérimenter le fait d'être « bien ensemble ».

À un niveau plus technique, on va aussi rapidement réaliser que l'information, par ce mécanisme de codage informatique, devient une matière abstraite qui se mesure.

Gérard Berry

« Un message, peu importe sa valeur réelle ou supposée, peu importe son sens exact ou erroné, contient une quantité précise d'information. L'atome d'information c'est l'élément binaire, le bit comme oui/non, 0/1, vrai/faux. Savoir de quelqu'un si c'est un homme ou une femme, un jeune ou un vieux, quelqu'un de grand ou petit, c'est très schématique mais cela nous donne déjà trois atomes d'informations sur lui, trois bits. La taille en information de deux informations indépendantes s'additionnent, mais pas celle de deux informations redondantes : par exemple si nous ajoutons que ce quelqu'un est un humain, on ne gagne rien, s'il est homme ou femme il est humain. Tous les objets : les images, les sons, les textes, les données ont un reflet numérique qui permet de mémoriser de l'information, de la transmettre, de la reproduire à l'infini. De la manipuler de manière spécifique aussi, grâce à des algorithmes. » [**5**]

Enfin, *last but not least*, nous avons fait la connaissance dans cette leçon de ***Grace Murray Hopper***, « la » pionnière de l'informatique.

À propos de son métier de mathématicienne et informaticienne, elle qui a aussi servi sa nation sous les drapeaux, voici ce qu'elle dit :

« Je me suis beaucoup amusée (I had a wonderful time), je suis vraiment chanceuse (I am really fortunate). »

Noter cependant que la **notion de compilateur** n'est qu'un prétexte ici et reste largement hors sujet.

Finalement, on a fait manipuler des objets « abstraits » à l'enfant. Certes. Mais pas à proprement parler des

objets « mathématiques ». C'est un autre domaine théorique qui est mis en jeu ici, une autre discipline abstraite : l'*informatique*.

P.S. :

La rédaction d'Images des maths et les auteurs remercient pour leur relecture attentive, les relecteurs Lebek, Bruno Duchesne, Mikaël Cabon et Maxime Bourrigan.

Notes

[1] Une version plus élaborée du jeu de Hamming est proposée par **Xavier Caruso** sous forme de **jeu web** dont nous réutilisons les éléments sous une forme très simplifiée ici.

[2] Dans un premier temps, on ne se préoccupe pas des nœuds papillons.

[3] Grâce à **Tralalère**, **Xprod**, **Inria**, avec **Universcience** et la plume d'**Audrey Mikaëlian**, une 20taine de pépites de science attachées à une personne qui a fait avancer la connaissance est racontée aux enfants : ce sont les **Sépas**. Contenu scientifique réalisé grâce à l'aide de **Nicolas Rougier**, **Sylvie Boldo**, et **Joanna Jongwane**

[4] Chanson Plus Bifluorée, « L'informatique »

[5] Gérard Berry est régent de Déformatique au **Collège de 'Pataphysique**.

Affiliation des auteurs

Thierry Viéville : Chercheur Inria, équipe mnemosyne. , **Aurélien Alvarez** : Université d'Orléans

---

Pour citer cet article : **Aurélien Alvarez** et **Thierry Viéville**, **« Dis papa (ou maman), comment arrivent les bugs dans le monde numérique ? »** — *Images des Mathématiques*, CNRS, 2014.

En ligne, URL : **http://images.math.cnrs.fr/Dis-papa-ou-maman-comment-arrivent.html**

Si vous avez aimé cet article, voici quelques suggestions automatiques qui pourraient vous intéresser :

- **Dis maman (ou papa), comment on cache des secrets dans le monde numérique ?**, par **Aurélien Alvarez** et **Thierry Viéville**
- **Dis maman (ou papa), c'est quoi un algorithme dans ce monde numérique ?**, par **Aurélien Alvarez** et **Thierry Viéville**
- **De l'ambiguïté des puzzles aux idées de Galois**, par **Xavier Caruso** et **Bruno Teheux**